



Second Generation Electronic Filing Specifications



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Court Filing 2.0 Version Test03

Staff and Consultants

Winchel "Todd" Vincent, III, Author
WTVIII, Inc. and <xmlLegal>
winchel@mindspring.com

Kevin Stannard, Editor
Swiftnet Solutions, Ltd.
kevin.stannard@swiftnet-solutions.com

Charlene Hammitt, Project Director
California Administrative Office of the Courts
charlene.hammitt@jud.ca.gov

Christopher Smith, Staff Analyst
California Administrative Office of the Courts
christopher.smith@jud.ca.gov

Tom Smith, Lead Consultant
AVI/ITDecisions
tjsmith@itdecision.com

Contributors

AVI/ITDecision

Tom Smith, tjsmith@itdecision.com

Counterclaim.com (<http://www.counterclaim.com>)

Shogan Naidoo shogan@counterclaim.com

Jim Beard beard@counterclaim.com

E-Filing.com (<http://www.e-filing.com>)

Mohammed Shaikh, mohammed@e-filing.com

Amrit Singh Nandrey, amrit@imagexx.com

Prabhath Pallati, prabhath@imagexx.com

Essential Publishers (<http://www.epubs.org>)

Martin Dean, mrsmary@epubs.org

ISD Corporation

Rob Beach, Ron.Beach@isd-corp.com

One Legal, Inc. (<http://www.onelegal.com>)

Robert DeFilippes, rtd@onelegal.com

Matt Marshall, matt@mattmarshall.com

Lexis-Nexis CourtLink (<http://www.lexisnexis.com>)

Shane Durham, shane.durham@lexisnexis.com

Sustain Technologies

Jon D. Salzman, jd.salzman@dailyjournal.com

Main Table of Contents

Filing Schema	3
Calendar Schema	14
Case Schema	18
Charge Schema	23
Confirmation Schema	33
CreditCard Schema	43
Document Schema	48
Envelope Schema	56
Header Schema	59
Page Schema	67
Payment Schema	70
Service Schema	72
Legal Notices	77



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Filing Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/>

Schema Prefix

Filing

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Filing.xsd>

Table of Contents

1. Elements

Internal

1.1. Filing	-4-
1.2. Date	-7-
1.3. Time	-7-
1.4. Filers	-7-
1.5. Filer	-7-
1.6. People	-7-
1.7. Organizations	-8-
1.8. Things	-8-
1.9. Charges	-8-
1.10. Fees	-8-
1.11. LeadDocuments	-9-
1.12. Attachments	-9-
1.13. Extensions	-9-
1.14. RunMode	-9-

External

1.15. Key	-9-
1.16. Case	-9-
1.17. CourtDetails	-9-
1.18. Person	-10-
1.19. Organization	-10-
1.20. Thing	-10-
1.21. Charge	-10-
1.22. Fee	-10-
1.23. Service	-10-
1.24. Coversheet	-10-
1.25. LeadDocument	-10-
1.26. Attachment	-11-
1.27. Payment	-11-
1.28. Calendar	-11-

1.29. Extension	-11-
2. Simple Types	
2.1. RunModes	-11-
3. Imported Schemas	
3.1. Attributes	-11-
3.2. CourtDetails	-11-
3.3. Extension	-11-
3.4. Fee	-12-
3.5. Key	-12-
3.6. Organization	-12-
3.7. Person	-12-
3.8. Thing	-12-
3.9. Calendar	-12-
3.10. Case	-12-
3.11. Charge	-12-
3.12. Document	-12-
3.13. Payment	-12-
3.14. Service	-12-
4. Change History	
4.1. 2003-02-18	-12-
4.2. 2003-03-01	-12-
4.3. 2003-06-14	-12-
4.4. 2003-06-14	-12-
4.5. 2003-06-14	-12-
4.6. 2003-06-15	-13-
4.7. 2003-07-23	-13-
4.8. 2003-07-26	-13-
4.9. 2003-07-27	-13-
4.10. 2003-07-29	-13-

1. Elements

1.1. Filing: Filing

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Key	1	1
Date	1	1
Time	1	1
Case	1	1
CourtDetails	1	1
Filers	1	1
People	0	1
Organizations	0	1
Things	0	1
Charges	0	1
Fees	0	1
Service	0	1

Coversheet	0	1
LeadDocuments	1	1
Attachments	0	1
Payment	0	1
Calendar	0	1
Extensions	0	1
RunMode	1	1

[1] This document is the [Court Filing 2.0 Specification](#). The web-based version is broken into several files, one for each schema. The Adobe PDF version is one PDF document for all schema unique to Filing and an additional PDF document that has documentation for "Building Block" schema. "Building Block" schema are common to both [Court Filing 2.0](#) and [Court Policy 2.0](#). Both the web-based format and the PDF format contain links to all schema documentation and the schema themselves. An accompanying document, [2GEFS Concepts](#), augments this specification with a high-level overview of terminology, models, and use cases.

[2] [Court Filing 2.0 XML \("CF 2.0"\)](#) includes requisite elements to capture information necessary to (a) initiate a case in a target Case Management System ("CMS"), if a case does not exist, (b) file one or more documents into a case, (c) send name, contact details, identifiers and descriptions, and roles for people and organizations associated with the case, (d) send charges or offenses in a case, (e) send and confirm service of process information (f) send and confirm payment information and (g) send and confirm calendar requests and schedules.

[3] [CF 2.0](#) filings and confirmations must be sent electronically to and from different systems. [CF 2.0](#) therefore includes (a) an XML transmission envelope (b) envelope header information and (c) filing confirmation XML. The [CF 2.0](#) transmission envelope can be substituted with a [SOAP 1.2](#) envelope. The [CF 2.0](#) header and confirmation XML can be used with [SOAP 1.2](#). There are, therefore, two similar means of transporting [CF 2.0](#) Filings and Confirmations.

[4] It is logically possible to transmit a Filing or a Confirmation by HTTP, HTTPS, FTP, SMTP, Facsimile, postal mail, and bar coded paper (fax or post). The recommended approach is to use HTTP, HTTPS and SMTP as the exclusive means for CF 2.0 transactions. Confirmations should be sent via HTTP, HTTPS, or SMTP based on the scenarios listed below and in the [2GEFS Concept](#) document. 2GEFS Interoperability Testing will test transmissions using these protocols only. SMTP transmissions must include an XML attachment in the notification or message in the email body.

[5] Below are five possible interchanges in which [CF 2.0](#) Filings and Confirmations can be used in an electronic filing system. These interchanges are illustrated in more detail in the [2GEFS Concepts](#) document.

[6] Filing can have two modes: (1) [CF1S: Filing: Synchronous](#) and (2) [CF1A: Filing: Asynchronous](#). [CF1A: Filing Asynchronous](#) is discouraged. Service Providers are not expected to provide an asynchronous means of electronic filing, although they may do so. 2GEFS Interoperability Testing will only test [CF1S: Filing Synchronous](#) over HTTP and HTTPS connections.

[7] Initial Confirmation can have two modes: (3) [CF2S: Initial Confirmation: Synchronous](#) and (4) [CF2A: Initial Confirmation: Asynchronous](#). [CF2A: Initial Confirmation: Asynchronous](#) is discouraged. Service providers are not expected to provide asynchronous means of sending confirmations, although they may do so.

[8] Filing applications are expected to send enveloped Filing XML to receiving applications over an HTTP connection initiated by the filing application.

[9] Receiving applications are expected to: (a) Receive the Filing. (b) Validate the Filing against the [CF 2.0](#) Filing Schema. (c) Programmatically check to ensure that at least one Plaintiff and one Defendant has been specified

and that the Court specified in the Filing is the correct Court. Other programmatic checking may be required. (d) Create an XML Confirmation and send it back to the sending application over the same HTTP connection as an HTTP response.

[10] Subsequent Confirmation can have one mode: (5) **CF3A: Subsequent Confirmation: Asynchronous**. Applications should send Subsequent Confirmations after the Filing transaction occurs at the appropriate point during the electronic acceptance/rejection process and CMS insertion process. Subsequent Confirmations can only be asynchronous. Subsequent Confirmations should be sent via HTTP to the sending application if the sending application has a means of accepting HTTP confirmations and has included a URI in the **Header:ReplyTo** element of the Filing Envelope Header. Subsequent Confirmations may also be sent via SMTP. SMTP Confirmations must include Confirmation XML as an attachment to the SMTP message.

[11] Simple username/password authentication information is included in **CF 2.0** Header. Additional Security or Authentication information may be provided using **SOAP 1.2** and related toolkits. Participants have agreed to the following rules regarding filer identification and authentication.

[12](a) At a minimum, Service Providers must identify Filers. Service Providers may also wish to authenticate the identity of Filers. Service Providers may identify or authenticate Filers using any means, including no means at all.

[13](b) The identification and authentication process for Filers should be as simple as possible with the fewest barriers balanced by reasonable security precautions.

[14](c) Any identification or authentication regime should be Service Provider agnostic. That is, filers should not be bound to a Service Provider by way of an identifier or authentication regime.

[15](d) Service Providers do not authenticate Filers on behalf of Courts, but must provide some type of Filer identification to the Court. Filer identification may be done by first and last name, bar number, social security number, federal taxpayer identification number, or other means acceptable to the court. Individual courts, on a case-by-case basis, may decide the type of identifier it requires.

[16](e) Service Providers must authenticate their software to the court. This can be done using the username/password information in the Header. To the extent it is feasible, service-provider-to-court authentication should be done using technology deemed to provide greater security than unencrypted username/passwords. However, service-provider-to-court authentication may be done by other means, such as accepting known filing URLs, IP filtering, or through relying on the barrier of knowing how to format an acceptable electronic filing and submit it to the court.

[17] For service-provider-to-court transactions, at a minimum, interoperability testing will include: (i) XML Over HTTP: XML over HTTP using Envelope with Filing inside. (ii) XML Over SHTTP: XML over SHTTP (SSL) using Envelope with Filing inside. Interoperability testing may also include (iii) Digitally Signed SOAP: XML over SHTTP (SSL) using SOAP wrapping Envelope with Filing inside.

[18] The **Filing** element is the intended root element of the schema.

[19] Filing must contain one **Key**, **Date**, **Time**, **Case**, **CourtDetails** and **Filers**.

[20] **Filing** must contain one **LeadDocuments/LeadDocument** element, but may contain several. This is not to say that there are more than one logical lead documents; rather, the purpose is to allow one lead document to be simultaneously filed in different electronic formats, such as Adobe PDF and TIFF. See **Filing:LeadDocument** and the **Document** Schema for more information.

[21] **Filing** may contain **People**, **Organizations**, **Things**, **Charges**, **Fees**, **Service**, **Coversheet**, **Attachments**, **Payment**, **Calendar**, and **Extensions**.

1.2. Filing: Date

Data Type: **xsd:date**

[22] **Filing:Date** is the date of the filing. The **Filing:Date** is not the same as the transmission date, although these date values may be the same or very close in time. **Filing:Date** is not a date from the document. **Filing:Date** is not the date of service of process.

1.3. Filing: Time

Data Type: **xsd:time**

[23] **Filing:Time** is the time of the filing. The **Filing:Time** is not the same as the transmission time, although these time values may be the same or very close in time.

1.4. Filing: Filers

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Filer	1	unbounded

[24] **Filing:Filers** is a container element for the **Filing:Filer** element. **Court Filing 1.0 ("CF 1.0")** does not define a standard means of specifying the filer. This ambiguity is a barrier to interoperability. **CF 2.0** uses the **Filing:Filers** to make the filer's information explicit. The filer's information should not be duplicated in **Filing:People** or **Filing:Organizations**.

1.5. Filing: Filer

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
Person	1	1
Organization	1	1

[25] **Filing:Filer** contains information about who filed the case. **Filing:Filer** may be a **Person** or an **Organization**. A person would be a natural person. An organization would be a legal entity, such as a corporation, a limited liability company, or a government subdivision.

[26] Specific courts and jurisdictions should define business rules for this element. There are a number of options for the meaning: (a) **Filing:Filer** could be the exact person who files the document, such as a legal secretary or a lawyer, (b) **Filing:Filer** could be a lawyer, even though the actual person who clicks the button and files the document is a lawyer's legal secretary, (c) **Filing:Filer** could be the firm for which a lawyer works, even though a lawyer or a lawyer's legal secretary actually clicks the button and files the document or (d) **Filing:Filer** could be the party on whose behalf a lawyer or law firm files a document. If it is unknown which of the above policies applies in a court, then if more than one **Filing:Filer** element exists, applications should consider the information in the first **Filing:Filer** element as the primary or only filer (depending on how many filers the application can accept). For example, the **CMS-API** specification (as of this writing) only allows one "FiledByParty" when adding a document to a CMS.

1.6. Filing: People

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Person	1	unbounded

[27] **Filing:People** is a container element for one or more **Filing:Person** elements. **Filing:People** are all of the people, whether parties, officials, or judges, associated with the case.

[28] On case initiation (e.g., a Filing without a **Case:Number**), this element should be populated with all of the parties, officials, and other people relevant to the case. This information should then be passed to the CMS via the **CMS-API** or **Request/Response XML**. In subsequent filings, this information should not be populated unless new people have been added to the case. Applications, using the **CMS-API**, should check to see whether people already exist in the CMS and should not insert people twice or should provide the clerk the opportunity to add, delete, or update people as appropriate.

1.7. Filing: Organizations

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Organization	1	unbounded

[29] **Filing:Organizations** is a container element for one or more **Filing:Organization** elements. **Filing:Organizations** are all of the organizations, whether parties, state agencies, or others, associated with the case.

[30] On case initiation (e.g., a Filing without a **Case:Number**), this element should be populated with all of the organizations relevant to the case. This information should then be passed to the CMS via the **CMS-API** or **Request/Response XML**. In subsequent filings, this information should not be populated unless new organizations have been added to the case. Applications, using the **CMS-API**, should check to see whether organizations already exist in the CMS and should not insert organizations twice or should provide the clerk the opportunity to add, delete, or update organizations as appropriate.

1.8. Filing: Things

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Thing	1	unbounded

[31] **Filing:Things** is a container element for one or more **Filing:Thing** elements. A **Filing:Thing** is an animal or inanimate object (i.e., not a legal entity) that has been made a party to a case.

[32] This is called "in rem" jurisdiction. In rem jurisdiction is sought in situations where a real person or organization (e.g., a legal entity) is not within the jurisdiction of the court and a plaintiff wants to gain financial or physical control over the property or thing. In this case, the "thing" being sued would be, for example, a boat, a car, a bank account, or a goat.

1.9. Filing: Charges

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Charge	1	unbounded

[33] **Filing:Charges** is a container element for one or more **Filing:Charge** elements. A **Filing:Charge** is the details of an offense or complaint in a criminal, juvenile, or other criminal-like case. **Filing:Charge** will not usually be used in civil cases.

1.10. Filing: Fees

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Fee	1	unbounded

[34] **Filing:Fees** is a container element for one or more **Filing:Fee** elements. See **Fee** Schema.

1.11. Filing: LeadDocuments

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
LeadDocument	1	4

[35] **Filing:LeadDocuments** is a container element for **Filing:LeadDocument**. A **Filing:LeadDocument** is the primary document. See **Filing:LeadDocument**.

1.12. Filing: Attachments

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Attachment	1	unbounded

[36] **Filing:Attachments** is a container element for **Filing:Attachment**. A **Filing:Attachment** is a secondary document, as described in **Filing:LeadDocument**. Other than its meaning, **Filing:Attachment** works in the same way as **Filing:LeadDocument**, except that **Filing:Attachment** can and should have different **Document:Key** values. See **Filing:LeadDocument** and **Document** Schema.

1.13. Filing: Extensions

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Extension	1	unbounded

[37] **Filing:Extensions** is a container element for **Filing:Extension**. A **Filing:Extension** is a generic name/value pair that allows applications to send information not included in the filing specification. See **Extension** Schema.

1.14. Filing: RunMode

Data Type: **RunModes**

[38] **Filing:RunMode** is a single element with two possible values: "Test" or "Live". RunMode is used when sending test filings to a live server so that downstream applications or the court clerk will know to ignore or behave differently toward the test filing. The element may also be useful in situations where test filings into a live server need to be purged mechanically, rather than deleted manually.

1.15. Filing: Key

Data Type: **Key:Key**

[39] **Filing:Key** is a unique identifier for the filing. See **Key** Schema.

1.16. Filing: Case

Data Type: **Case:Case**

[40] **Filing:Case** contains information about the case being filed. This information is primarily used for case initiation. See **Case** Schema.

1.17. Filing: CourtDetails

Data Type: **CourtDetails: CourtDetails**

[41] **Filing: CourtDetails** contains information about the court into which documents are being filed. See **CourtDetails** Schema.

1.18. Filing: Person

Data Type: **Person: Person**

[42] **Filing: Person** contains information about a person. See **Filing: People** documentation. See **Person** Schema.

1.19. Filing: Organization

Data Type: **Organization: Organization**

[43] **Filing: Organization** contains information about a organization. See **Filing: Organizations** documentation. See **Organization** Schema.

1.20. Filing: Thing

Data Type: **Thing: Thing**

[44] **Filing: Thing** contains information about a thing. See **Filing: Things** documentation. See **Thing** Schema.

1.21. Filing: Charge

Data Type: **Charge: Charge**

[45] **Filing: Charge** contains information about a charge. See **Filing: Charges** documentation. See **Thing** Schema.

1.22. Filing: Fee

Data Type: **Fee: Fee**

[46] **Filing: Fee** contains information about a court filing fee. See **Fee** Schema and Court Policy 2.0 Specification.

1.23. Filing: Service

Data Type: **Service: Service**

[47] **Filing: Service** contains information about documents that have been processed served. See **Service** Schema.

1.24. Filing: Coversheet

Data Type: **Document: Document**

[48] **Filing: Coversheet** contains a filing coversheet document, if any. The actual document should be included in the **Document: Pages** element. Associated data, if any, should be put in the **Document: DataFile** element. **Filing: Coversheet** is not a lead document and is not an attachment.

1.25. Filing: LeadDocument

Data Type: **Document: Document**

[49] **Filing: LeadDocument** is the primary and most important document in a filing. For example, if a party is filing a Motion for Summary Judgment supported by an Affidavit and two Exhibits, then the Motion would be the lead document and the other three documents would be attachments.

[50] It is possible to have more than one **Filing: LeadDocument** elements although there can only be one logical lead document. The ability to use multiple **Filing: LeadDocument** elements is so that the same document can be transmitted in several different electronic formats, such as Microsoft Word and Adobe PDF. There is an arbitrary limit of four **Filing: LeadDocument** elements that can be used as it is unlikely that more than four formats of the

same document will be sent to a court.

[51]Multiple **Filing:LeadDocument** elements should not be used to send data files associated with a document. For example, it is not permissible to send a Microsoft Word document in one **Filing:LeadDocument** along with an associated data file containing data within the Microsoft Word document in another **Filing:LeadDocument**. It is possible to send a document and associated data file using the **Document:DataFile** element.

[52]The **Document:Key** for each **Filing:LeadDocument** must be the same. Applications that receive different **Document:Key** values should use the value from the first **Filing:LeadDocument**. See **Document** Schema.

1.26. Filing: Attachment

Data Type: **Document:Document**

[53] **Filing:Attachment** contains information about a secondary document. See **Filing:LeadDocument** and **Filing:Attachments**.

1.27. Filing: Payment

Data Type: **Payment:Payment**

[54]A **Filing:Payment** provides information about a filer's credit card so that a filer can pay a service provider or the court directly. See **Payment** Schema.

1.28. Filing: Calendar

Data Type: **Calendar:Calendar**

[55]A **Filing:Calendar** provides information about a calendared events. See **Calendar** Schema.

1.29. Filing: Extension

Data Type: **Extension:Extension**

[56]A **Filing:Extension** is a generic name/value pair that allows applications to send information not included in the filing specification. See **Filing:Extensions**.

2. Simple Types

2.1. RunModes

Data Type: **xsd:string**

Enumeration(s)
Test
Live

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. CourtDetails

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/CourtDetails/01/>

3.3. Extension

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Extension/01/>

3.4. Fee

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Fee/01/>

3.5. Key

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Key/01/>

3.6. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

3.7. Person

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Person/01/>

3.8. Thing

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Thing/01/>

3.9. Calendar

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Calendar/01/>

3.10. Case

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Case/01/>

3.11. Charge

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Charge/01/>

3.12. Document

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Document/01/>

3.13. Payment

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Payment/01/>

3.14. Service

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Service/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-03-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/02/>

4.3. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Test01/>

Copied. Removed extra min and maxOccurs.

4.4. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test01/>

Copied. Changed People content model from choice to sequence.

4.5. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test02/>

Copied. Added RunMode element, complexType and documentation. Added RunModes simpleType. Created container element, LeadDocuments.

4.6. 2003-06-15

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/>

Copied. Eliminated Date.xsd and Time.xsd and included xsd:date and xsd:time. Added Payment Schema.

4.8. 2003-07-26

Editor: Winchel Vincent

Added Calendar schema. Changed Key, CourtDetails, and Extension namespaces and import locations to reflect move of schema to Building Block folder.

4.9. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Changed Person and Organization namespaces.

4.10. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Calendar Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Calendar/01/>

Schema Prefix

Calendar

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Calendar/01/Calendar.xsd>

Table of Contents

1. Elements

Internal

1.1. Calendar	-14-
1.2. CaseTitle	-15-
1.3. CaseNumber	-15-
1.4. CourtDepartmentNumber	-15-
1.5. Date	-15-
1.6. Events	-15-
1.7. Event	-15-
1.8. Time	-15-
1.9. Duration	-16-
1.10. Description	-16-
1.11. StatutoryPriority	-16-
1.12. Status	-16-

External

2. Simple Types

2.1. Statuses	-16-
2.2. Memo	-16-

3. Imported Schemas

3.1. Attributes	-16-
------------------------	------

4. Change History

4.1. 2003-07-26	-17-
4.2. 2003-07-27	-17-
4.3. 2003-07-29	-17-

1. Elements

1.1. Calendar: Calendar

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
CourtDepartmentNumber	0	1
Date	1	1
Events	1	1

[1] **Calendar** is the intended root element of the schema. **Calendar** contains information about events on a Calendar. A Calendar represents one day.

1.2. Calendar: CaseTitle

Data Type: **xsd:string**

[2] **Calendar:CaseTitle** is the title or style of the case for a event. For example, **Jones v. Harris**.

1.3. Calendar: CaseNumber

Data Type: **xsd:string**

[3] **Calendar:CaseNumber** is the case number of the case for an event.

1.4. Calendar: CourtDepartmentNumber

Data Type: **xsd:string**

[4] **Calendar:CourtDepartmentNumber** is a the department number of the court.

1.5. Calendar: Date

Data Type: **xsd:date**

[5] **Calendar:Date** is the date on which the calendared events take place.

1.6. Calendar: Events

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Event	1	unbounded

[6] **Calendar:Events** is a container element for one more more **Calendar:Event** elements. **Calendar:Events** are events that take place on the day of the Calendar.

1.7. Calendar: Event

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
CaseTitle	1	1
CaseNumber	1	1
Time	1	1
Duration	0	1
Description	0	1
StatutoryPriority	0	1
Status	0	1

[7] **Calendar:Event** is an event that takes place on the day of the Calendar.

1.8. Calendar: Time

Data Type: **xsd:time**

[8] **Calendar:Time** is the start time of a calendared event.

1.9. Calendar: Duration

Data Type: **xsd:string**

[9] **Calendar:Duration** is the duration of a calendared event.

1.10. Calendar: Description

Data Type: **Memo**

[10] **Calendar:Description** is a description of a calendared event.

1.11. Calendar: StatutoryPriority

Data Type: **xsd:boolean**

[11] **Calendar:StatutoryPriority** is a boolean value that, when true, is a representation by a filer that the filer believes the event has a statutory priority to be calendared before other events.

1.12. Calendar: Status

Data Type: **Statuses**

[12] **Calendar:Status** is the status of a calendared event. **Calendar:Status** can have one of the following values (a) **Request** (b) **Scheduled** (c) **Rescheduled** (d) **RequestRejected** (e) **Unknown**. When a calendar event is sent in a Filing, the calendar event should usually be a **Request** for the event to be calendared. For example, a request for a hearing date. When a calendar event is sent in a confirmation, the event should usually be a confirmation that the event has been **Scheduled** on the calendar, but it could also be a **RequestRejected**. A subsequent confirmation may also show that a calendared event has been **Rescheduled**.

2. Simple Types

2.1. Statuses

Data Type: **xsd:string**

Enumeration(s)
Request
Scheduled
Rescheduled
RequestRejected
Unknown

2.2. Memo

Data Type: **xsd:string**

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

4. Change History

4.1. 2003-07-26

Editor: Winchel Vincent

Created.

4.2. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.3. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Calendar/01/>

Copied. Added StatutoryPriority element.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Case Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Case/01/>

Schema Prefix

Case

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Case/01/Case.xsd>

Table of Contents

1. Elements

Internal

1.1. Case	-19-
1.2. New	-19-
1.3. Existing	-19-
1.4. Number	-19-
1.5. Type	-19-
1.6. Category	-20-
1.7. Title	-20-
1.8. NumberOfParties	-20-
1.9. UnnamedParties	-20-
1.10. AmountInControversy	-20-
1.11. Year	-20-

External

2. Simple Types

3. Imported Schemas

3.1. Attributes	-20-
-----------------	------

4. Change History

4.1. 2003-02-18	-21-
4.2. 2003-03-01	-21-
4.3. 2003-06-14	-21-
4.4. 2003-06-14	-21-
4.5. 2003-06-14	-21-
4.6. 2003-06-15	-21-
4.7. 2003-07-23	-21-
4.8. 2003-07-25	-21-
4.9. 2003-07-27	-22-
4.10. 2003-07-29	-22-

1. Elements

1.1. Case: Case

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
New	1	1
Existing	1	1

[1] **Case:Case** is the intended root element of the schema.

1.2. Case: New

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Number	0	1
Type	1	1
Category	1	1
Title	1	1
NumberOfParties	0	1
UnnamedParties	0	1
AmountInControversy	0	1
Year	0	1

[2] **Case:New** is .

1.3. Case: Existing

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Number	1	1
Type	0	1
Category	0	1
Title	0	1
NumberOfParties	0	1
UnnamedParties	0	1
AmountInControversy	0	1
Year	0	1

[3] **Case:Existing** is .

1.4. Case: Number

Data Type: **xsd:string**

[4] **Case:Number** is the case number or unique identifier into which documents are to be filed. If **Case:Number** is blank, then **Case:New** should be set to "Yes". Regardless of the value in **Case:New**, if the **Case:Number** is not present, then applications should assume that the case is a case initiation or should return an error. Whether this error results in a rejected filing is a policy that should be set by the court.

1.5. Case: Type

Data Type: **xsd:string**

[5] **Case:Type** is the type of new case initiated. This element should only be used on case initiation. It distinguishes, for example, a new case that is created as a result of a complaint, an appeal, or a transfer from a

local or foreign jurisdiction. If **Court Policy XML** exists for the court, this value should come from the Case Filing Type or similar table in **Court Policy XML**.

1.6. Case: Category

Data Type: **xsd:string**

[6] **Case:Category** is the court case category associated with the case. If **Court Policy XML** exists for the court, this value should come from the Case Category or similar table in **Court Policy XML**.

1.7. Case: Title

Data Type: **xsd:string**

[7] **Case:Title** is the title or style of the case. In all or nearly all jurisdictions, the rule for creating a case title (also called a caption) is to create a string using the last name of the first plaintiff listed in the title of the complaint, followed by " v. ", followed by the last name of the first defendant listed in the caption on the complaint. If either party is an organization (e.g., legal entity), then the name of the organization is used. For example, "IBM, Inc." or "State of Georgia." Once created, the **Case:Title** does not change (i.e., the Case:Title should only be sent to the CMS on case initiation.

[8]The receiving application must not accept the value sent in **Case:Title**. It may generate its own case title internally. The **Case:Title** should be sent back in the Confirmation XML.

1.8. Case: NumberOfParties

Data Type: **xsd:integer**

[9] **Case:NumberOfParties** is the total number of parties named in the filing.

1.9. Case: UnnamedParties

Data Type: **xsd:integer**

[10] **Case:UnnamedParties** is the number of parties that are associated with the case, but not named in the filing.

1.10. Case: AmountInControversy

Data Type: **xsd:string**

[11] **Case:AmountInControversy** is the amount in controversy. The currency is assumed to be the currency of the court's jurisdiction.

1.11. Case: Year

Data Type: **xsd:string**

[12] **Case:Year** is the year in which the case is being filed.

2. Simple Types

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-03-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/02/Case/01/>

Added NumberOfParties, UnnamedParties and AmountInControversy elements. Add Type element.

4.3. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Test01/Case/01/>

Copied. Eliminated UnnamedPartiesBegin and End elements.

4.4. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test01/Case/01/>

Copied

4.5. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test02/Case/01/>

Copied

4.6. 2003-06-15

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Case/01/>

Copied. Changed "FilingType" to "Type". Added new "New" and "Existing" elements. Changed Case content model to choice of New and Existing. Eliminated old New element. Eliminated YesNo simpleType used on old New element. Changed required/optional qualities of elements to reflect required/optional information for a new/existing case.

4.8. 2003-07-25

Editor: Winchel Vincent

Changed Caption element to Title element to be consistent with new Calendar schema.

4.9. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.10. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Case/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Charge Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Charge/01/>

Schema Prefix

Charge

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Charge/01/Charge.xsd>

Table of Contents

1. Elements

Internal

1.1. Charge	-24-
1.2. Name	-24-
1.3. StateCode	-24-
1.4. NCICCode	-24-
1.5. NCICFreeText	-24-
1.6. Type	-24-
1.7. Date	-25-
1.8. DateRangeBegin	-25-
1.9. DateRangeEnd	-25-
1.10. Time	-25-
1.11. TimeVariation	-25-
1.12. Description	-25-

External

1.13. Location	-25-
----------------	------

2. Simple Types

2.1. NCICCodes	-25-
2.2. Types	-30-
2.3. TimeVariations	-30-

3. Imported Schemas

3.1. Attributes	-31-
3.2. Address	-31-

4. Change History

4.1. 2003-02-18	-32-
4.2. 2003-03-01	-32-
4.3. 2003-06-14	-32-
4.4. 2003-06-14	-32-
4.5. 2003-06-14	-32-
4.6. 2003-06-15	-32-
4.7. 2003-07-23	-32-

4.8. 2003-07-27
4.9. 2003-07-29

-32-
-32-

1. Elements

1.1. Charge: Charge

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Name	1	1
StateCode	0	1
NCICCode	0	1
NCICFreeText	0	1
Type	0	1
Date	0	1
Time	0	1
TimeVariation	0	1
Location	0	1
Description	0	1

[1] Charge is the intended root element of the schema.

1.2. Charge: Name

Data Type: xsd:string

[2] Charge:Name is the name of the charge, sometimes defined by statute. For example, 'Battery' is the name of a charge.

1.3. Charge: StateCode

Data Type: xsd:string

[3] Charge:StateCode is the statutory or other code section under which the offender is charged. For example, 'O.C.G.A. 16-5-23.1' is the code section for juvenile Battery in Georgia.

1.4. Charge: NCICCode

Data Type: NCICCodes

[4] Charge:NCICCode are NCIC Codes defined as national standards.

1.5. Charge: NCICFreeText

Data Type: xsd:string

[5] Charge:NCICFreeText is a place where text can entered to modify an NCIC free text code.

1.6. Charge: Type

Data Type: Types

[6] Charge:Type is the type of charge, in this case 'felony' or 'misdemeanor.'

[7] **ISSUE: There are currently only two values allowed for this element: "Misdemeanor" and "Felony."**

Should there be other values, such as "Class A Misdemeanor" or "Class B Misdemeanor". Alternatively, any value could be allowed.

1.7. Charge: Date

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
DateRangeBegin	1	1
DateRangeEnd	1	1

[8] Charge:Date is a container element for Charge:DateRangeBegin and Charge:DateRangeEnd.

1.8. Charge: DateRangeBegin

Data Type: xsd:date

[9] Charge:DateRangeBegin is the first known date on which an offense began. If Charge:DateRangeBegin and Charge:DateRangeEnd have the same value, then that value is the exact date of the offense.

1.9. Charge: DateRangeEnd

Data Type: xsd:date

[10] Charge:DateRangeEnd is the last known date of the offense. If Charge:DateRangeBegin and Charge:DateRangeEnd have the same value, then that value is the exact date of the offense.

1.10. Charge: Time

Data Type: xsd:time

[11] Charge:Time is the exact or estimated time of the offense. Charge:Time can be qualified by Charge:TimeVariation.

1.11. Charge: TimeVariation

Data Type: TimeVariations

[12] Charge:TimeVariation is an enumeration of plus/minus time values that allow Charge:Time to be qualified. For example, if the time of the offense can be estimated to be 4:00 p.m. +/- 1 hour.

1.12. Charge: Description

Data Type: xsd:string

[13] Charge:Description is a description of the offense.

1.13. Charge: Location

Data Type: Address:Address

[14] Charge:Location is the location of the offense, expressed as an Address.

2. Simple Types

2.1. NCICCodes

Data Type: xsd:string

Enumeration(s)
0101 [Treason]

0103 [Espionage]
0104 [Sabotage]
0105 [Sedition]
0106 [Selective Service]
0199 [Sovereignty Free Text]
0201 [Desertion]
0299 [Military Offense Free Text]
0301 [Illegal Entry]
0302 [False Citizenship]
0303 [Smuggling Aliens]
0399 [Immigration Free Text]
0901 [Willful Killing - Family - Gun]
0902 [Willful Killing - Family - Other Weapon]
0903 [Willful Killing - NonFamily - Gun]
0904 [Willful Killing - NonFamily - Other Weapon]
0905 [Willful Killing - Public Officer - Gun]
0906 [Willful Killing - Public Officer - Other Weapon]
0907 [Willful Killing - Police Officer - Gun]
0908 [Willful Killing - Police Officer - Other Weapon]
0909 [Negligent Manslaughter - Vehicle]
0910 [Negligent Manslaughter - Gun/Other Weapon]
0911 [Willful Killing - Gun]
0912 [Willful Killing - Other Weapon]
0999 [Homicide Free Text]
1005 [Kidnapping - Minor]
1006 [Kidnapping - Adult]
1009 [Hijack - Aircraft]
1099 [Kidnapping FreeText]
1101 [Rape - Gun]
1102 [Rape - Other Weapon]
1103 [Rape - Strongarm]
1104 [Sodomy]
1116 [Statutory Rape]
1199 [Sexual Assault Free Text]
1201 [Robbery - Business - Gun]
1202 [Robbery - Business - Other Weapon]
1203 [Robbery - Business - Strongarm]
1204 [Robbery - Street - Gun]
1205 [Robbery - Street - Other Weapon]
1206 [Robbery - Street - Strongarm]
1207 [Robbery - Residence - Gun]
1208 [Robbery - Residence - Other Weapon]
1209 [Robbery - Residence - Strongarm]
1210 [Forcible Purse Snatching]
1211 [Robbery - Bank/Financial Institution]
1299 [Robbery FreeText]
1313 [Simple Assault/Simple Battery]
1314 [Aggravated Assault - Gun]

1315 [Aggravated Assault - Other Weapon]
1316 [Intimidation/Terroristic Threats]
1399 [Assault Fee Text]
1499 [Abortion Free Text]
2005 [Arson - Business]
2006 [Arson - Residence]
2009 [Arson - Public Building]
2099 [Arson - FreeText]
2101 [Extortion - Threat To Injure Person]
2102 [Extortion - Threat To Damage Property]
2199 [Extortion Free Text]
2202 [Burglary - Forced Entry - Residence]
2203 [Burglary - Forced Entry - NonResidence]
2204 [Burglary - No Forced Entry - Residence]
2205 [Burglary - Forced Entry - NonResidence]
2206 [Possession of Burglary Tools]
2299 [Burglary Free Text]
2301 [Pocket Picking]
2302 [Larceny - Sudden Snatch]
2303 [Shoplifting]
2304 [Larceny - Parts From Vehicle]
2305 [Larceny - From Vehicle]
2307 [Larceny - From Coin Machine]
2308 [Larceny - From Building]
2309 [Larceny - From Yards]
2310 [Larceny - From Mail]
2314 [Larceny - Government Property]
2316 [Larceny - From Residential Mailbox]
2317 [Larceny - Bicycle]
2318 [Larceny - Livestock]
2374 [Larceny - Gas Drive Off]
2399 [Larceny Free Text]
2404 [Theft of Automobile]
2406 [Theft By Receiving Stolen Vehicle]
2407 [Stripping Stolen Vehicle]
2408 [Possession of Stolen Vehicle]
2410 [Theft of Aircraft]
2424 [Theft of Truck or Bus]
2434 [Theft of Other Vehicle]
2499 [Stolen Vehicle Free Text]
2501 [Forgery of Checks]
2502 [Forgery of Other Objects]
2589 [Forgery Free Text]
2599 [Counterfeiting Free Text]
2602 [Fraud - Swindle]
2604 [Fraud - Impersonation]
2605 [Fraud - Illegal Use of Credit Cards]
2606 [Fraud - Bad/Worthless Checks]

2699 [Fraudulent Activity Free Text]
2701 [Embezzlement - Business Property]
2703 [Embezzlement - Financial Institution]
2799 [Embezzlement - Free Text]
2711 [Conversion of Leased Personal Property]
2801 [Sale of Stolen Property]
2802 [Transportation Interstate of Stolen Property]
2803 [Theft By Receiving Stolen Property]
2804 [Possession of Stolen Property]
2899 [Stolen Property Free Text]
2901 [Damage To Property - Business]
2902 [Damage To Property - Private]
2903 [Damage To Property - Public]
2999 [Damage To Property Free Text]
2921 [Criminal Interference w/Govt. Property]
3501 [Hallucinogen - Manufacturing]
3502 [Hallucinogen - Distribution Of]
3503 [Hallucinogen - Sales Of]
3504 [Hallucinogen - Possession Of]
3505 [Hallucinogen Free Text]
3510 [Heroin - Sales Of]
3511 [Heroin - Smuggling]
3512 [Heroin - Possession Of]
3513 [Heroin Free Text]
3520 [Opium or Derivative - Sales Of]
3521 [Opium or Derivative - Smuggling]
3522 [Opium or Derivative - Possession Of]
3523 [Opium or Derivative Free Text]
3530 [Cocaine - Sales Of]
3531 [Cocaine - Smuggling]
3532 [Cocaine - Possession Of]
3533 [Cocaine Free Text]
3540 [Synthetic Narcotic - Sales Of]
3541 [Synthetic Narcotic - Smuggling]
3542 [Synthetic Narcotic - Possession Of]
3543 [Synthetic Narcotic Free Text]
3550 [Possession of Drug Related Items/Narcotic Equipment]
3560 [Marijuana - Sales Of]
3561 [Marijuana - Smuggling]
3562 [Marijuana - Possession Of]
3563 [Marijuana - Manufacturing]
3564 [Marijuana Free Text]
3570 [Amphetamine - Manufacturing]
3571 [Amphetamine - Sales Of]
3572 [Amphetamine - Possession Of]
3573 [Amphetamine Free Text]
3580 [Barbiturate - Manufacturing]
3581 [Barbiturate - Sales Of]

3582 [Barbiturate - Possession Of]
3583 [Barbiturate Free Text]
3599 [Dangerous Drugs Free Text]
3601 [Child Molestation]
3604 [Incest]
3605 [Indecent Exposure]
3606 [Bestiality]
3608 [Adultery]
3611 [Peeping Tom]
3699 [Sex Offense Free Text]
3707 [Obscene Communication]
3799 [Obscenity Free Text]
3801 [Family Neglect]
3802 [Cruelty Toward Child]
3804 [Bigamy]
3805 [Contributing To The Delinquency of A Minor]
3806 [Child Neglect]
3899 [Family Offense Free Text]
3902 [Operating A Card Game]
3904 [Playing In A Card Game]
3905 [Operating A Dice Game]
3907 [Playing In A Dice Game]
3921 [Establishing A Gambling Place]
3999 [Gambling Free Text]
3932 [Commercial Gambling]
4001 [Keeping A House of Ill Repute]
4002 [Procuring For Prostitution (Pimping)]
4004 [Prostitution]
4099 [Commercialized Sex Free Text]
4101 [Liquor - Manufacturing]
4102 [Liquor - Sales Of]
4103 [Liquor - Transportation Of]
4104 [Liquor - Possession Of]
4105 [Misrepresentation of Age]
4199 [Liquor Law Violation Free Text]
4299 [Drunkenness]
4801 [Resisting Officer]
4802 [Obstructing Criminal Investigation]
4803 [Making A False Report]
4805 [Tampering With A Witness]
4807 [Refusing To Aid An Officer]
4812 [Failure To Report A Crime]
4899 [Obstructing Police Free Text]
4859 [Fleeing/Attempting To Elude]
4901 [Escape]
4902 [Flight To Avoid Prosecution/Confinement]
4903 [Aiding Prisoner To Escape]
4904 [Harboring An Escapee or Fugitive]

4999 [Flight/Escape Free Text]
5003 [Perjury]
5005 [Contempt of Court]
5006 [Obstructing Justice]
5007 [Obstructing Court Order]
5011 [Parole Violation]
5012 [Probation Violation]
5015 [Failure To Appear]
5099 [Obstructing Judiciary Free Text]
5007IWC [Interference w/Custody]
5009REV [Revocation of Probation]
5102 [Bribe - Offering]
5103 [Bribe - Receiving]
5109 [Gratuity Free Text]
5199 [Bribery Free Text]
5201 [Weapon - Altering The Identification Of]
5202 [Carrying A Concealed Weapon]
5203 [Carrying A Prohibited Weapon]
5210 [Weapon - Licensing/Registration]
5211 [Possession of Explosives]
5212 [Possession Of Other Weapon]
5213 [Firing A Weapon]
5215 [Bomb Threat]
5216 [Arson Threat]
5299 [Weapons Offense Free Text]
5254 [Possession of Firearm/Knife During Commission of Crime]
5308 [False Fire Alarm]
5309 [Harassing Communication]
5311 [Disorderly Conduct]
5312 [Disturbing The Peace]
5312 [Curfew Violation]
5314 [Loitering Violation]
5399 [Public Peace Free Text]
5365 [Reckless Conduct]
5403 [Driving Under The Influence of Drugs]
5404 [Driving Under The Influence of Alcohol]
5412 [Driving with License Suspended/Revoked]
5499 [Other Traffic Offenses Free Text]
5599 [Health/Safety Free Text]
5699 [Civil Rights Free Text]

2.2. Types

Data Type: **xsd:string**

Enumeration(s)
Felony
Misdemeanor

2.3. TimeVariations

Data Type: **xsd:string**

Enumeration(s)
+/- 5 minutes
+/- 10 minutes
+/- 15 minutes
+/- 20 minutes
+/- 25 minutes
+/- 30 minutes
+/- 35 minutes
+/- 40 minutes
+/- 45 minutes
+/- 50 minutes
+/- 55 minutes
+/- 1 hour
+/- 2 hours
+/- 3 hours
+/- 4 hours
+/- 5 hours
+/- 6 hours
+/- 7 hours
+/- 8 hours
+/- 9 hours
+/- 10 hours
+/- 11 hours
+/- 12 hours
+/- 13 hours
+/- 14 hours
+/- 15 hours
+/- 16 hours
+/- 17 hours
+/- 18 hours
+/- 19 hours
+/- 20 hours
+/- 21 hours
+/- 22 hours
+/- 23 hours

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Address

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Address/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-03-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/02/Charge/01/>

4.3. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Test01/Charge/01/>

Copied

4.4. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test01/Charge/01/>

Copied

4.5. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test02/Charge/01/>

Copied

4.6. 2003-06-15

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Charge/01/>

Copied. Changed Time.xsd to xsd:time. Changed Location from xsd:string to Address.xsd.

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.9. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Charge/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Confirmation Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Confirmation/01/>

Schema Prefix

Confirmation

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Confirmation/01/Confirmation.xsd>

Table of Contents

1. Elements

Internal

1.1. Confirmation	-34-
1.2. Number	-35-
1.3. Date	-35-
1.4. Time	-35-
1.5. FilingStatus	-35-
1.6. ErrorMessage	-36-
1.7. FilingDate	-36-
1.8. FilingTime	-36-
1.9. Filers	-37-
1.10. Filer	-37-
1.11. People	-37-
1.12. Organizations	-37-
1.13. Things	-37-
1.14. Charges	-37-
1.15. Fees	-38-
1.16. LeadDocuments	-38-
1.17. Attachments	-38-
1.18. Extensions	-38-
1.19. RunMode	-38-

External

1.20. FilingKey	-39-
1.21. Case	-39-
1.22. CourtDetails	-39-
1.23. Person	-39-
1.24. Organization	-39-
1.25. Thing	-39-
1.26. Charge	-39-
1.27. Fee	-39-
1.28. Service	-39-

1.29. Coversheet	-39-
1.30. LeadDocument	-40-
1.31. Attachment	-40-
1.32. Payment	-40-
1.33. Calendar	-40-
1.34. Extension	-40-
2. Simple Types	
2.1. FilingStatuses	-40-
2.2. RunModes	-40-
3. Imported Schemas	
3.1. Attributes	-41-
3.2. CourtDetails	-41-
3.3. Extension	-41-
3.4. Fee	-41-
3.5. Key	-41-
3.6. Organization	-41-
3.7. Person	-41-
3.8. Thing	-41-
3.9. Calendar	-41-
3.10. Case	-41-
3.11. Charge	-41-
3.12. Document	-41-
3.13. Payment	-41-
3.14. Service	-41-
4. Change History	
4.1. 2003-01-15	-41-
4.2. 2003-07-21	-41-
4.3. 2003-07-21	-41-
4.4. 2003-07-23	-42-
4.5. 2003-07-26	-42-
4.6. 2003-07-27	-42-
4.7. 2003-07-27	-42-
4.8. 2003-07-29	-42-

1. Elements

1.1. Confirmation: Confirmation

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Number	0	1
Date	1	1
Time	1	1
FilingStatus	1	1
ErrorMessage	0	1
FilingKey	0	1
FilingDate	0	1
FilingTime	0	1

Case	0	1
CourtDetails	0	1
Filers	0	1
People	0	1
Organizations	0	1
Things	0	1
Charges	0	1
Fees	0	1
Service	0	1
Coversheet	0	1
LeadDocuments	0	1
Attachments	0	1
Payment	0	1
Calendar	0	1
Extensions	0	1
RunMode	1	1

[1]The **Confirmation** element is the intended root element of the schema. The Confirmation mirrors the elements in a Filing, except that a Confirmation also includes a **Confirmation:Number**, a **Confirmation:Date**, a **Confirmation:Time**, a **Confirmation:FilingStatus**, and a **Confirmation:ErrorMessage** (if any). Elements in the Confirmation mirror elements in a Filing based on the assumption that Filing information may change as the Filing XML travels downstream from a filing application to the court. In case the information changes, or even if it does not, the Confirmation serves as a receipt and notice of the information ultimately accepted by the court. As a result, there is a corresponding element in the Confirmation for every element in the Filing. Applications are not required to mirror all elements and return them in a Confirmation; however, applications may do so.

[2]For example, the filing application may include an incorrect or incomplete value for a case category. The clerk of court or the EFM itself may add or change the case category value. In this case, the application creating the Confirmation may include the value in the Confirmation in the same element in the XML as in the Filing and send the Confirmation to appropriate addresses.

1.2. Confirmation: Number

Data Type: **xsd:string**

[3] **Confirmation:Number** is a sequential number generated by a receiving application as a receipt for a Filing. Delivery of a **Confirmation:Number** value to a sending application from a receiving application signifies that the sending application no longer has responsibility to file or re-file the document.

1.3. Confirmation: Date

Data Type: **xsd:date**

[4] **Confirmation:Date** is the date on which the Confirmation is sent.

1.4. Confirmation: Time

Data Type: **xsd:time**

[5] **Confirmation:Time** is the Time on which the Confirmation is sent.

1.5. Confirmation: FilingStatus

Data Type: **FilingStatuses**

[6] **Confirmation:FilingStatus** is the status of the filing. **Confirmation:FilingStatus** can have the following values (a)

Received, (b) Accepted, (c) Pending, (d) Rejected, and (e) Error.

[7]The value **Received** means that, at least, the receiving application has checked that the Filing XML is well-formed and valid against the schema. Receiving applications may also programmatically check whether certain values in the XML are logically correct. For example, a receiving application may check to ensure that Plaintiff and Defendant names are acceptable values or that all fees have been paid. If Filing XML is acceptable to the receiving application, then the receiving application must return the value **Received** (or other acceptable value as defined in this section) along with a **Confirmation:Number**. Sending the value **Received** with a **Confirmation:Number** means that the filing application no longer has a responsibility to file or re-file.

[8]The value **Accepted** means that the filing has been accepted by the court as filed or lodged. (It is recognized that terms such as "filed" and "lodged" and their meanings may differ in different jurisdiction. Jurisdictions should define the meaning of **Accepted** in applicable court rules.) The **Confirmation:FilingDate** and **Confirmation:FilingTime** are the date and time when the court accepted the filing. The **Confirmation:FilingDate** and **Confirmation:FilingTime** values may or may not be the same as the **Confirmation:Date** and **Confirmation:Time**. A **Confirmation:Number** should be sent with the **Accepted** status.

[9]The value **Pending** means that the filing is still pending as of the **Confirmation:Date** and **Confirmation:Time**. A **Confirmation:Number** should be sent with the **Pending** status.

[10]The value **Rejected** means that the filing has been rejected by the court at a time subsequent to the initial receipt. It is, therefore, possible that a Filer could receive a **Confirmation:Number** and a **Received** status, but that the document might later be rejected. Court Rules should address this possibility. A **Confirmation:Number** should be sent with the **Rejected** status.

[11]The value **Error** means that there has been a technical error and the Filing has not been **Received**. If the XML is not well-formed and valid, then the receiving application must return an **Error**. If the errors in the Filing XML are so severe that the receiving application cannot respond with a Confirmation, then the filing application should generate a Confirmation. A Confirmation with a **Confirmation:FilingStatus** value equal to **Error** should also include a **Confirmation:ErrorMessage** with a value describing the error.

[12]It is important to note that the primary purpose of the **Confirmation:FilingStatus** element and its values is to provide a standard and interoperable technical infrastructure that can support different court policies or rules. Individual courts or entire jurisdictions are encouraged to promulgate rules that support the **Confirmation:FilingStatus** values.

1.6. Confirmation: ErrorMessage

Data Type: **xsd:string**

[13] **Confirmation:ErrorMessage** is descriptive text about an error used when **Confirmation:FilingStatus** value is **Error**.

1.7. Confirmation: FilingDate

Data Type: **xsd:date**

[14] **Confirmation:FilingDate** is the date of the filing as confirmed by the court. The **Confirmation:FilingDate** is not the same as the transmission date, although these date values may be the same or very close in time. **Confirmation:FilingDate** is not a date from the document. **Confirmation:FilingDate** is not the date of service of process.

1.8. Confirmation: FilingTime

Data Type: **xsd:time**

[15] **Confirmation:FilingTime** is the time of the filing. The **Confirmation:FilingTime** is not the same as the transmission time, although these time values may be the same or very close in time.

1.9. Confirmation: Filers

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Filer	1	unbounded

[16] **Confirmation:Filers** is a container element for the **Confirmation:Filer** element.

1.10. Confirmation: Filer

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
Person	1	1
Organization	1	1

[17] **Confirmation:Filer** contains information about who filed the case as accepted by the court. See **Person** and **Organization** Schemas.

1.11. Confirmation: People

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Person	1	unbounded

[18] **Confirmation:People** is a container element for one or more **Confirmation:Person** elements. **Confirmation:People** should include people added to the case by the court. See **Person** Schema.

1.12. Confirmation: Organizations

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Organization	1	unbounded

[19] **Confirmation:Organizations** is a container element for one or more **Confirmation:Organization** elements. **Confirmation:Organizations** should include organizations added to the case by the court. See **Organization** Schema.

1.13. Confirmation: Things

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Thing	1	unbounded

[20] **Confirmation:Things** is a container element for one or more **Confirmation:Thing** elements. **Confirmation:Things** should include things added to the case by the court. See **Thing** Schema.

1.14. Confirmation: Charges

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Charge	1	unbounded

[21] **Confirmation:Charges** is a container element for one or more **Confirmation:Charge** elements. **Confirmation:Charges** should include charges added to the case by the court. See **Charge** Schema.

1.15. Confirmation: Fees

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Fee	1	unbounded

[22] **Confirmation:Fees** is a container element for one or more **Confirmation:Fee** elements. **Confirmation:Fees** should include fees charged to the filer by the court. See **Fee** Schema.

1.16. Confirmation: LeadDocuments

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
LeadDocument	1	4

[23] **Confirmation:LeadDocuments** is a container element for **Confirmation:LeadDocument**. **Confirmation:LeadDocuments** should include lead documents added to the case by the court. There is no requirement to send back the base64 encoded document, although applications may do so. Applications are instead encouraged to send back a link to the filed document. See **Document** Schema.

1.17. Confirmation: Attachments

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Attachment	1	unbounded

[24] **Confirmation:Attachments** is a container element for **Confirmation:Attachment**. **Confirmation:Attachments** should include attachments added to the case by the court. There is no requirement to send back the base64 encoded document, although applications may do so. Applications are instead encouraged to send back a link to the filed document(s). See **Document** Schema.

1.18. Confirmation: Extensions

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Extension	1	unbounded

[25] **Confirmation:Extensions** is a container element for **Confirmation:Extension**. In a filing, a **Filing:Extension** is a generic name/value pair that allows filing applications to send information not included in the filing specification. The filing extension may have a attribute value equal to **Yes**. Responding applications that encounter a value equal to **Yes** are required to send back the exact name/value pair in **Confirmation:Extension**. See **Extension** Schema.

1.19. Confirmation: RunMode

Data Type: **RunModes**

[26] **Confirmation:RunMode** is a single element with two possible values: **Test** or **Live**. **Filing:RunMode** is used when sending test filings to a live server so that downstream applications or the court clerk will know to ignore or behave differently toward the test filing. The element may also be useful in situations where test filings into a live server need to be purged mechanically, rather than deleted manually. **Confirmation:RunMode** must mirror the value sent in the original filing.

1.20. Confirmation: FilingKey	Data Type: Key:Key
[27] Confirmation:FilingKey is the value of the Filing:Key in the original filing.	
1.21. Confirmation: Case	Data Type: Case:Case
[28] Confirmation:Case contains information about the case filed as confirmed by the court. See Case Schema.	
1.22. Confirmation: CourtDetails	Data Type: CourtDetails:CourtDetails
[29] Confirmation:CourtDetails contains information about the court into which documents has been filed as confirmed by the court. See CourtDetails Schema.	
1.23. Confirmation: Person	Data Type: Person:Person
[30] Confirmation:Person contains information about a person. See Confirmation:People documentation. See Person Schema.	
1.24. Confirmation: Organization	Data Type: Organization:Organization
[31] Confirmation:Organization contains information about a organization. See Confirmation:Organizations documentation. See Organization Schema.	
1.25. Confirmation: Thing	Data Type: Thing:Thing
[32] Confirmation:Thing contains information about a thing. See Confirmation:Things documentation. See Thing Schema.	
1.26. Confirmation: Charge	Data Type: Charge:Charge
[33] Confirmation:Charge contains information about a charge. See Confirmation:Charges documentation. See Charge Schema.	
1.27. Confirmation: Fee	Data Type: Fee:Fee
[34] Confirmation:Fee contains information about a court filing fee. See Confirmation:Fees documentation. See Fee Schema and Court Policy 2.0 specification.	
1.28. Confirmation: Service	Data Type: Service:Service
[35] Confirmation:Service contains information about documents that have been processed served. See Service Schema.	
1.29. Confirmation: Coversheet	Data Type: Document:Document

[36] **Confirmation:Coversheet** contains a filing coversheet document, if any, as accepted by the court. There is no requirement to send back the base64 encoded document, although applications may do so. Applications are instead encouraged to send back a link to the filed document. See **Document** Schema.

1.30. Confirmation: LeadDocument

Data Type: **Document:Document**

[37] **Confirmation:LeadDocument** is the primary and most important document in a filing as accepted by the court. See **Confirmation:LeadDocuments** documentation and **Document** Schema.

1.31. Confirmation: Attachment

Data Type: **Document:Document**

[38] **Confirmation:Attachment** contains information about a secondary documents as accepted by the court. See **Confirmation:LeadDocument** and **Confirmation:Attachments**.

1.32. Confirmation: Payment

Data Type: **Payment:Payment**

[39]A **Confirmation:Payment** provides confirmation information about a payment made by a filer. In a **Confirmation:Payment** only the **Payment:Total** should be included as confirmation of the total amount charged to the filer's credit card. See **Payment**Schema.

1.33. Confirmation: Calendar

Data Type: **Calendar:Calendar**

[40]A **Confirmation:Calendar** is provides confirmation information about calendared events. See **Calendar** Schema.

1.34. Confirmation: Extension

Data Type: **Extension:Extension**

[41]A **Confirmation:Extension** is is a generic name/value pair that allows applications to send information not included in the confirmation specification. See **Confirmation:Extensions**.

2. Simple Types

2.1. FilingStatuses

Data Type: **xsd:string**

Enumeration(s)
Received
Accepted
Pending
Rejected
Error

2.2. RunModes

Data Type: **xsd:string**

Enumeration(s)
Test

Live

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. CourtDetails

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/CourtDetails/01/>

3.3. Extension

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Extension/01/>

3.4. Fee

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Fee/01/>

3.5. Key

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Key/01/>

3.6. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

3.7. Person

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Person/01/>

3.8. Thing

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Thing/01/>

3.9. Calendar

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Calendar/01/>

3.10. Case

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Case/01/>

3.11. Charge

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Charge/01/>

3.12. Document

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Document/01/>

3.13. Payment

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Payment/01/>

3.14. Service

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Service/01/>

4. Change History

4.1. 2003-01-15

Editor: Winchel Vincent

Initial Draft

4.2. 2003-07-21

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.3. 2003-07-21

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Confirmation/01/>

Copied

4.4. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Confirmation/01/>

Copied

4.5. 2003-07-26

Editor: Winchel Vincent

Added new Payment and Calendar Schemas.

4.6. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Deleted Date.xsd and Time.xsd. Changed Key, CourtDetails, and Extension namespaces and import locations to reflect move of schema to Building Block folder.

4.7. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Changed Person and Organization namespaces.

4.8. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Confirmation/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

CreditCard Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/CreditCard/01/>

Schema Prefix

CreditCard

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/CreditCard/01/CreditCard.xsd>

Table of Contents

1. Elements

Internal

1.1. CreditCard	-44-
1.2. Number	-44-
1.3. Expiration	-44-
1.4. Day	-44-
1.5. Month	-44-
1.6. Year	-44-
1.7. SecurityCode	-44-
1.8. Network	-44-
1.9. Encrypted	-45-

External

1.10. CardHolder	-45-
1.11. Bank	-45-

2. Simple Types

2.1. Days	-45-
2.2. Months	-46-
2.3. Years	-46-
2.4. SecurityCodes	-46-

3. Imported Schemas

3.1. Attributes	-46-
3.2. Organization	-46-
3.3. Person	-47-

4. Change History

4.1. 2003-MM-DD	-47-
4.2. 2003-07-27	-47-
4.3. 2003-07-29	-47-

1. Elements

1.1. CreditCard: CreditCard

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Number	1	1
Expiration	1	1
SecurityCode	0	1
CardHolder	1	1
Network	1	1
Bank	0	1
Encrypted	1	1

[1] **CreditCard** is the intended root element of the schema. **CreditCard** contains information used for making a credit card payment.

1.2. CreditCard: Number

Data Type: **xsd:string**

[2] **CreditCard:Number** is the sixteen-digit credit card number on the face of the card. Spaces are not allowed.

1.3. CreditCard: Expiration

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Day	0	1
Month	1	1
Year	1	1

[3] **CreditCard:Expiration** is the date and month the card expires.

1.4. CreditCard: Day

Data Type: **Days**

[4] **CreditCard:Day** is the day.

1.5. CreditCard: Month

Data Type: **Months**

[5] **CreditCard:Month** is the month the credit card expires represented as a two-digit number.

1.6. CreditCard: Year

Data Type: **Years**

[6] **CreditCard:Year** is the year the credit card expires represented as a two-digit number.

1.7. CreditCard: SecurityCode

Data Type: **SecurityCodes**

[7] **CreditCard:SecurityCode** is a three-digit security code located on the back of a credit card on or near the customer's signature.

1.8. CreditCard: Network

Data Type: **xsd:string**

[8] **CreditCard:Network** is the credit card network, such as Visa or MasterCard.

1.9. CreditCard: Encrypted

Data Type: **xsd:boolean**

[9]The **CreditCard:Encrypted** element is a boolean value that indicates, when true, that the element values for **CreditCard:Number**, **CreditCard:Expiration**, **CreditCard:SecurityCode** and are encrypted and, when false, that the same elements are unencrypted. For security reasons, the encryption algorithm and the encryption key are not included. It is assumed in a single-key cryptosystem that the parties exchanging information will know the encryption algorithm and the encryption key.

[10]The purpose of encrypting the elements is to avoid sending and storing the information in the clear. It must be stressed, however, that when using a single-key cryptosystem the sending party, the receiving party, or any other party that knows or can discover the encryption algorithm and the encryption key could easily forge, discover, or later manipulate the information.

[11]Applications wishing to use two-key cryptosystems or public key infrastructure should use the **W3C XML Signature** specification.

1.10. CreditCard: CardHolder

Data Type: **Person:Person**

[12] **CreditCard:CardHolder** is name and billing address of the person who holds the card.

1.11. CreditCard: Bank

Data Type: **Organization:Organization**

[13] **CreditCard:Bank** includes name and contact details of the bank that owns the card.

2. Simple Types

2.1. Days

Data Type: **xsd:string**

Enumeration(s)
01
02
03
04
05
06
07
08
09
10
11
12
13
14

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

2.2. Months

Data Type: **xsd:string**

Enumeration(s)
01
02
03
04
05
06
07
08
09
10
11
12

2.3. Years

Data Type: **xsd:string**
 maxLength: 2
 minLength: 2

2.4. SecurityCodes

Data Type: **xsd:string**
 maxLength: 3
 minLength: 3

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

3.3. Person

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Person/01/>

4. Change History

4.1. 2003-MM-DD

Editor: Winchel Vincent

Added structured Copyright and Update History.

4.2. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Changed Person and Organization namespaces.

4.3. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/CreditCard/01/>

Copied. Changed Numbers to xsd:string. Added Day element.



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Document Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Document/01/>

Schema Prefix

Document

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Document/01/Document.xsd>

Table of Contents

1. Elements

Internal

1.1. Document	-49-
1.2. Key	-49-
1.3. Title	-49-
1.4. Type	-50-
1.5. SpecificType	-50-
1.6. Code	-50-
1.7. Dates	-50-
1.8. Date	-50-
1.9. Times	-51-
1.10. Time	-51-
1.11. File	-51-
1.12. DataFile	-51-
1.13. MIMEType	-52-
1.14. FileName	-52-
1.15. FileExtension	-52-
1.16. Hash	-52-
1.17. Extensions	-53-

External

1.18. Page	-53-
1.19. Extension	-53-

2. Simple Types

2.1. Encodings	-53-
2.2. MIMETypes	-53-
2.3. Algorithms	-54-

3. Imported Schemas

3.1. Attributes	-54-
3.2. Extension	-54-
3.3. Page	-54-

4. Change History

4.1.	2003-02-18	-54-
4.2.	2003-03-01	-54-
4.3.	2003-06-14	-54-
4.4.	2003-06-14	-54-
4.5.	2003-06-14	-54-
4.6.	2003-06-16	-55-
4.7.	2003-07-23	-55-
4.8.	2003-07-27	-55-
4.9.	2003-07-29	-55-

1. Elements

1.1. Document: Document

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Key	1	1
Title	0	1
Type	0	1
SpecificType	0	1
Code	0	1
Dates	0	1
Times	0	1
File	0	1
DataFile	0	1
MIMEType	1	1
FileName	0	1
FileExtension	0	1
Hash	0	1
Extensions	0	1

[1] **Document** is the intended root element of the schema.

1.2. Document: Key

Data Type: **xsd:string**

[2] **Document:Key** is a unique identifier for the document. However, there is no requirement that the **Document:Key** value be unique within the filing. It is possible, for example, to send the same document, but in different electronic formats, such as Adobe PDF, TIFF, Microsoft Word, etc., in the same filing by creating multiple Document elements with the same **Document:Key** value. If there are multiple **Filing:LeadDocument** elements, then each element should all have the same **Document:Key** value.

1.3. Document: Title

Data Type: **xsd:string**

[3] **Document:Title** is the title of the document as it appears on the face of the document. **Document:Title** is not the same as **Document:Type** or **Document:Filename**. Many developers confuse **Document:Title** with the electronic document's computer filename. An additional element is included in this schema for **Document:Filename**. **Document:Filename** element is not necessary needed by the CMS, but may be useful in coordinating filenames and should help to distinguish between a **Document:Title** and a **Document:Filename**.

[4] For images or for documents that are not structured, it is difficult, perhaps impossible to know the [Document:Title](#) without looking at the face of the document. Accordingly, it is suggested that applications provide a user interface for the filer to type the [Document:Title](#) manually. See [Document:Type](#) and [Document:Filename](#).

1.4. Document: Type

Data Type: [xsd:string](#)

[5] [Document:Type](#) is a value usually found in a CMS code table and used by the court to categorize types of documents. A court's [Court Policy XML](#) should include a table with [Document:Type](#) values and this element should be populated with a value from this list. [Document:Type](#) is not the [Document:Title](#), although the values may be similar or the same. Usually, the [Document:Type](#) is more general than [Document:Title](#). [Document:Type](#) is not the type of electronic file (e.g., it should be distinguished from [Document:MIMETYPE](#)). See [Document:Title](#) and [Document:Code](#).

1.5. Document: SpecificType

Data Type: [xsd:string](#)

[6] [Document:SpecificType](#) is a general document type value that is used by vendors to match and categorize more specific [Document:Type](#) values used in a CMS.

[7] Suggested [Document:Specific](#) types are Affidavit (a) Answer (b) Brief (c) Certificate of Service (d) Complaint (e) Withdrawal Of Counsel (f) Correspondence (g) Counterclaim (h) Crossclaim (i) Demand for Jury Trial (j) Demand for Speedy Trial (k) Discovery (l) Dismissal (m) Exhibit (n) Interpleader (o) Judgment (p) Jury List (q) Motion (r) Notice (s) Order (t) Sentence (u) Sheriff Entry Of Service: Not Served (v) Sheriff Entry Of Service: Served (w) Subpoena (x) Transcript (y) Verdict (z) Verification (aa) Warrant. The values are specified in [Court Policy XML](#) code tables under the key DocumentSpecificType. These values may be varied by vendors or courts, by agreement of parties involved in exchanging information.

[8] **ISSUE: Do we want to hardcode these values into a simpleType for use in California.**

1.6. Document: Code

Data Type: [xsd:string](#)

[9] [Document:Code](#) is a code value usually found in a CMS code table used by the court to categorize types of documents. A CMS document code is usually associated with human-readable document type value, represented in this schema as [Document:Type](#). See [Document:Type](#).

1.7. Document: Dates

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Date	1	unbounded

[10] [Document:Dates](#) are any important dates on the face of the document, such as a hearing date. The type of date should be specified in the global attribute attribute.

1.8. Document: Date

Data Type: [xsd:date](#)

[11] [Document:Date](#) is a date on the face of the document. A single document may have many dates on it. [Document:Date](#) should be qualified by the global attribute (e.g., SignatureDate, AuthoredOnDate, EffectiveDate, HearingDate). [Document:Date](#) is not used by most CMSs. [Document:Date](#) may be the same as [Service:Date](#) for process of service; however, applications should not rely on [Document:Date](#) for the date of service of a document. See [Service](#) Schema.

1.9. Document: Times

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Time	1	unbounded

[12] **Document:Times** are any important times shown on the face of the document. Most documents will not have a **Document:Time** and this value is usually not needed by the CMS. The type of time should be specified in the global attribute attribute.

1.10. Document: Time

Data Type: **xsd:time**

[13] **Document:Time** is a time on the face of the document. **Document:Time** should be qualified by the global attribute (e.g., HearingTime). Most documents will not have a **Document:Time** and this value is usually not needed by the CMS.

1.11. Document: File

Content Model: sequence

Mixed: true

ChildElement(s)	minOccurs	maxOccurs
Page	0	unbounded

Attribute(s)	type	use	fixed/default
Encoding	Encodings	optional	None

[14] **Document:File** is the element in which an entire encoded document is included, whether or not the document is split into multiple files for each page. **Document:File** is set to "mixed" content. This means that either (a) the entire encoded document can be included in the **Document:File** element as a text node or (b) **Document:File** may act as a container element for a document that has pages separated into separate separate electronic files.

[15] In the latter case, **Document:File** must have one or more **Document:Page** elements as children and each page must be encoded and included as a text node of the **Page:Content** element of the **Page** schema in order (e.g., page one is encoded in the **Document:Page** first child; page two is encoded in **Document:Page** second child, etc.). There is no requirement to split documents into pages and, indeed, this practice is discouraged. (However, electronic files should always be split into separate documents, e.g., five logically separate documents should not be included in one electronic file.) Individual **Document:Page** elements should only be used in situations where (a) the electronic document is already split into separate File and it would be difficult to combine them or (b) the court requests the document come in separate File.

[16] **Document:File** has an attribute Encoding with a simpleType Encodings. Encodings has three values (a) Base64 (b) Link and (c) DIME. Base64 encoding means that the electronic document is Base64 encoded and then encoded text placed in Document:File or Document:Page element. Base64 encoding is the preferred means of sending documents in Filing XML in situations where the total size of the Filing XML is under 2.5 megabytes. Link means that the electronic document is available outside of the Filing XML via a link (to a URL or a local drive) included in the Document:File or Document:Page elements. Link is useful in situations where the total size of the Filing XML would be over 2.5 megabytes. Link should not be used unless necessary. DIME is a recent Microsoft Note posted in the W3C's website of interest to several 2GEFS Participants. DIME is not a W3C standard as of this writing. The DIME attribute value is available as a placeholder for potential future use should the DIME specification mature to a recommended W3C standard.

1.12. Document: DataFile

Data Type: **xsd:string**

[17] **Document:DataFile** is a data file in any format (e.g., delimited text, xml, etc.), encoded as specified in the **Document:Encoding** element, that is associated with the document

1.13. Document: MIMEType

Data Type: **MIMETypes**

[18] **Document:MIMEType** is one of the standard **Document:MIMETypes**. A court or jurisdiction may limit the types of documents permitted to be filed in the court.

[19] **ISSUE: Presently a limited number of MIME Types are hard coded in an enumeration. This has been done because people often forget the MIME Types or get them wrong. This also forces the developer to use the MIME Type. The disadvantage of this approach is that if a MIME Type is needed that is not in the list, it will not be available.**

1.14. Document: FileName

Data Type: **xsd:string**

[20] **Document:FileName** is the filename of the electronic file. If the document is split into File, then the string "_XXX" should be appended to the end of each filename corresponding to each page, in order, where "XXX" is a zero padded number starting with "_001". For example, if there are three File and the **Document:FileName** value is 00018001, then the three resulting filenames should be 00018001_001, 00018001_002, and 00018001_003. The file extension should not be included in **Document:FileName**.

1.15. Document: FileExtension

Data Type: **xsd:string**

[21] **Document:FileExtension** is the file extension associated with the document. The file extension should not be included in **Document:FileName**.

1.16. Document: Hash

Data Type: **xsd:string**

[22] **Document:Hash** is a hash value of the document. The **Document:Hash** has an attribute that may be one of four values (a) **SHA-1** (b) **SHA-256** (c) **SHA-384** (d) **SHA-512**.

[23] This element has been added as a result of a requirement in **Standards for Electronic Filing Processes (February 26th, 2003)** to use FIPS 180-2 for message integrity. FIPS 180-2 states "This standard specifies four secure hash algorithms, SHA-1, SHA-256, SHA-384, and SHA-512. All four of the algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest. These algorithms enable the determination of a message's integrity: any change to the message will, with a very high probability, result in a different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes, and in the generation of random numbers (bits)."

[24] Contrary to assertions made in **Standards for Electronic Filing Processes (February 26th, 2003)** courts and service providers using hashes without digital signatures should understand that storage of many hashes and/or sending a hash back to a filer in a confirmation message is not a foolproof method of assuring message integrity. Without digital signatures, it is possible for parties on either side of a transaction to alter a document, produce a new hash value, and claim that the new document is the authentic document using the new hash as proof. In such a case, there could be a dispute as to which party is actually in possession of the authentic document. Because either a filer or a filer's staff or a court or a court's staff could be the malefactor, there can be no presumption, using a hash function alone, that one side or the other is telling the truth. Additional security procedures are required.

[25] Applications wishing to use two-key cryptosystems or public key infrastructure should use the [W3C XML Signature](#) specification.

1.17. Document: Extensions

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Extension	1	unbounded

[26] [Document:Extensions](#) is a container element for [Document:Extension](#). A [Document:Extension](#) is a generic name/value pair that allows applications to send information not included in the document schema. See [Extension](#) Schema.

1.18. Document: Page

Data Type: [Page:Page](#)

[27] [Document:Page](#) element contains the content of a page encoded as specified in the [Document:Encoding](#) element. See [Page](#) Schema.

1.19. Document: Extension

Data Type: [Extension:Extension](#)

[28] A [Document:Extension](#) is a generic name/value pair that allows applications to send information not included in the document schema. See [Extension](#) Schema.

2. Simple Types

2.1. Encodings

Data Type: [xsd:string](#)

Enumeration(s)
Base64
Link
DIME

2.2. MIMETypes

Data Type: [xsd:string](#)

Enumeration(s)
text/html
text/plain
text/richtext
text/xml
image/tiff
image/gif
image/jpeg
image/bmp
application/msword
application/octet-stream
application/pdf
Other

2.3. Algorithms

Data Type: **xsd:string**

Enumeration(s)
SHA-1
SHA-256
SHA-384
SHA-512

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Extension

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Extension/01/>

3.3. Page

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Page/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-03-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/02/Document/01/>

Added SpecificType element. Added Dates and Times elements as container elements for Date and Time.

4.3. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Test01/Document/01/>

Copied. Removed mixed content from Dates and Times.

4.4. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test01/Document/01/>

Copied

4.5. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test02/Document/01/>

Copied

4.6. 2003-06-16

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Added text/xml MIME Type.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Document/01/>

Copied. Changed Date.xsd and Time.xsd to xsd:date and xsd:time. Eliminated Encoding element and added Encoding attribute and Encodings simpleType. Added Hash element and Algorithm attribute. Added Extensions and Extension elements and Extension.xsd.

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.9. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Document/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Envelope Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Envelope/01/>

Schema Prefix

Envelope

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Envelope/01/Envelope.xsd>

Table of Contents

1. Elements

Internal

- 1.1. **Envelope** -56-
- 1.2. **Body** -57-

External

- 1.3. **Header** -57-

2. Simple Types

3. Imported Schemas

- 3.1. **Attributes** -57-
- 3.2. **Filing** -57-
- 3.3. **Confirmation** -57-
- 3.4. **Header** -57-

4. Change History

- 4.1. **2003-01-15** -57-
- 4.2. **2003-07-21** -57-
- 4.3. **2003-07-21** -57-
- 4.4. **2003-07-23** -58-
- 4.5. **2003-07-27** -58-
- 4.6. **2003-07-29** -58-

1. Elements

1.1. Envelope: Envelope

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Header	1	1
Body	1	1

[1] **Envelope** is the intended root element of the schema. An **Envelope** is a container element for a

Envelope:Header and an **Envelope:Body**.

1.2. Envelope: Body
Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
Filing:Filing	1	1
Confirmation:Confirmation	1	1

[2] **Envelope:Body** contains information necessary for the primary transaction. In this case, **Envelope:Body** may contain either a **Filing** or a **Confirmation**.

1.3. Envelope: Header
Data Type: Header:Header

[3] **Envelope:Header** contains transmission information necessary for moving data in the **Envelope:Body** from one address to another address. Once a message has reached its final destination, the **Envelope:Header** may be retained for logging purposes, but it otherwise unnecessary for the primary transaction.

2. Simple Types

3. Imported Schemas

3.1. Attributes
http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/
3.2. Filing
http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/
3.3. Confirmation
http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Confirmation/01/
3.4. Header
http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Header/01/

4. Change History

4.1. 2003-01-15
Editor: Winchel Vincent

Initial Draft

4.2. 2003-07-21
Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.3. 2003-07-21
Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Envelope/01/>

Copied

4.4. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Envelope/01/>

Copied

4.5. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.6. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Envelope/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Header Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Header/01/>

Schema Prefix

Header

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Header/01/Header.xsd>

Table of Contents

1. Elements

Internal

1.1. Header	-60-
1.2. To	-60-
1.3. CC	-61-
1.4. From	-61-
1.5. ReplyTo	-62-
1.6. URIs	-62-
1.7. People	-62-
1.8. Organizations	-62-
1.9. URI	-63-
1.10. Credentials	-63-
1.11. Username	-63-
1.12. Password	-63-
1.13. Encrypted	-63-
1.14. MessageIdentifiers	-64-
1.15. MessageIdentifier	-64-
1.16. ConfirmationIdentification	-64-
1.17. Date	-64-
1.18. Time	-65-
1.19. Extensions	-65-

External

1.20. Person	-65-
1.21. Organization	-65-
1.22. Extension	-65-

2. Simple Types

3. Imported Schemas

3.1. Attributes	-66-
3.2. Extension	-66-
3.3. Organization	-66-
3.4. Person	-66-

4. Change History

4.1.	2003-01-15	-66-
4.2.	2003-07-21	-66-
4.3.	2003-07-21	-66-
4.4.	2003-07-23	-66-
4.5.	2003-07-26	-66-
4.6.	2003-07-27	-66-
4.7.	2003-07-29	-66-

1. Elements

1.1. Header: Header

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
To	0	1
CC	0	1
From	0	1
ReplyTo	0	1
Credentials	1	1
MessageIdentifiers	1	1
Date	1	1
Time	1	1
Extensions	0	1

[1] **Header** is the intended root element of the schema. **Header** contains information useful for transmitting Filing XML and Confirmation XML contained in a corresponding **Body** element. **Header** information is not needed to accomplish a filing. However, certain information, such a **Header:ReplyTo** and **Header:MessageIdentifier**, is necessary to send asynchronous confirmations. If there is no need to send asynchronous confirmations, then Header information may be discarded or it may be stored for logging purposes. Note that Filing XML includes a globally unique **Filing:Key**, so filings can be tracked in this manner.

[2]The **Header** may be used with an xmlLegal Envelope or with a **SOAP 1.2** Envelope. The **SOAP** specification explicitly states that application-specific header information is not defined by the **SOAP** specification and should be defined by the application. If **SOAP** is used for transmission, then the xmlLegal Header should go inside the **SOAP** Header element as the first child element. **SOAP** attributes designed to tell SOAP applications whether to understand or forward Header information may be used based on the rules defined for each element in this specification.

[3]The advantage of using an xmlLegal Envelope is that the Envelope, Header, and either the Filing or Confirmation, whichever is present, can be constructed and validated as a whole simply with a validating parser. There is no need to learn, use, or build a SOAP toolkit or to interoperate with SOAP created by other SOAP toolkits. The disadvantage of using an xmlLegal Envelope is that it only supports encrypted username/password authentication using single-key cryptosystem. Developers wishing to implement two-key cryptosystems or public key infrastructure are encouraged to use **SOAP 1.2** and associated toolkits that implement the **W3C XML Signature** specification.

1.2. Header: To

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
-----------------	-----------	-----------

URIs	1	1
------	---	---

[4] **Header:To** includes URIs to which the filing is being or has been sent. For example, if a filing application sends an Envelope with a Header and a Filing in it to a URI, then the URI value should be included in the **Header:URI** element that is a child to the **Header:To/Header:URIs** element (e.g., **Header:To/Header:URIs/Header:URI**). The receiving application does not need to understand the **Header:To/Header:URIs/Header:URI** element, since the receiving application is at the location of the URI. However, this information is useful for logging purposes.

[5] **Header:To** information must be forwarded by an intermediary application until it reaches the court. Intermediary application must add new **Header:To/Header:URIs/Header:URI** elements as the first child of the **Header:URIs** element each time the Filing XML is forwarded.

[6]The Filing XML should only be sent to one **Header:To/Header:URIs/Header:URI** at a time. If the Filing XML is sent to more than one URI, then the sending application must ensure that if there are any intermediary applications, only one application forwards the Filing on to the court. If this rule is not imposed, then the same Filing XML could be filed twice. It is suggested that the the primary or forwarding URI be sent in the **Header:To/Header:URIs/Header:URI** element and any secondary or non-forwarding URIs be sent in **Header:CC/Header:URIs/Header:URI** elements.

[7]For example, suppose a filing application creates a Filing XML and sends it to (a) a service provider's intermediary server at <http://www.one.com/> and (b) a lawyer's server at <http://www.two.com/>. The intention is for the service provider to forward the Filing XML on to the Court's server and for the lawyer's server merely to receive notice that the filing had been made. In this case, the value <http://www.one.com/> would go in **Header:To/Header:URIs/Header:URI** and the value <http://www.two.com/> would go in **Header:CC/Header:URIs/Header:URI**.

[8]The filing may also be CCed to people and organizations via email and facsimile in the **Header:CC** element. See **Header:CC**, **Header:Person** and **Header:Organization**.

1.3. Header: CC

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
URIs	1	1
People	1	1
Organizations	1	1

[9] **Header:CC** includes addresses to which the filing is being or has been sent. **Header:CC** works in the same way as **Header:To** with respect to URIs, except that inclusion of a URI in **Header:CC** means that the application at the URI is not meant to forward the Filing XML to the court.

[10]Intermediary applications should understand **Header:CC** information to ensure that Filings are not forwarded after receipt. Applications may, but are not required to, forward **Header:CC** information.

[11]The Filing XML can be CCed to people and organizations via email and facsimile. See **Header:Person** and **Header:Organization**.

1.4. Header: From

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
People	1	1

Organizations	1	1
---------------	---	---

[12] **Header:From** includes the names and contact details of people and organizations from which the filing is being or has been sent.

[13] **Header:From** does not include means of specifying a URI because a URI will never be a sending application. However, it is possible to specify a URI in the **Header:ReplyTo** element. In the case of an asynchronous transaction, for example, a response, such as Confirmation XML, would be sent to the URIs in **Header:ReplyTo** element. In the case of a synchronous transaction, a response would be sent in the HTTP response over the same HTTP connection as the original message, which does not require a URI.

[14]Receiving applications do not have to understand information in the **Header:From** element, but may wish to present such information to users or store it for logging purposes. Intermediary applications must forward **Header:From** information to the court or other final destination.

1.5. Header: ReplyTo

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
URIs	1	1
People	1	1
Organizations	1	1

[15] **Header:ReplyTo** contains addresses or facsimile numbers to which asynchronous replies should be sent. Receiving applications must understand and send replies to URIs and email addresses in the **Header:ReplyTo** element. Messages sent to email addresses should include a text or HTML message in the email body and XML as an email attachment. Receiving applications are not required to understand or send replies to facsimile numbers or postal addresses, but may do so. Sending applications must not include any address or number of any kind unless a reply is meant to be sent to the address or number.

[16]If intermediary applications are involved, then each application that wishes to receive a reply upon a downstream event must add an address in **Header:ReplyTo**.

1.6. Header: URIs

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
URI	1	unbounded

[17] **Header:URIs** is a container element for one or more **Header:URI** elements.

1.7. Header: People

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Person	1	unbounded

[18] **Header:People** is a container element for one or more **Header:Person** elements.

1.8. Header: Organizations

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
-----------------	-----------	-----------

Organization	1	unbounded
--------------	---	-----------

[19] **Header:Organizations** is a container element for one or more **Header:Organization** elements.

1.9. Header: URI

Data Type: **xsd:string**

[20] **Header:URI** contains a address to which a filing or confirmation is being or has been sent.

1.10. Header: Credentials

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Username	1	1
Password	1	1
Encrypted	1	1

[21] **Header:Credentials** is a container element for username and password information.

1.11. Header: Username

Data Type: **xsd:string**

[22] **Header:Username** is the username of the person or application sending the **Filing** or **Confirmation**. If the **Header:Encrypted** element is set to false, then the **Header:Username** value must be unencrypted. If the **Header:Encrypted** element is set to true, then the **Header:Username** value must be encrypted.

1.12. Header: Password

Data Type: **xsd:string**

[23] **Header:Password** is the password of the person or application sending the **Filing** or **Confirmation**. If the **Header:Encrypted** element is set to false, then the **Header:Password** value must be unencrypted. If the **Header:Encrypted** element is set to true, then the **Header:Password** value must be encrypted.

1.13. Header: Encrypted

Data Type: **xsd:boolean**

[24]The **Header:Encrypted** element is a boolean value that indicates, when true, that the **Header:Username** and **Header:Password** are encrypted and, when false, that the **Header:Username** and **Header:Password** are unencrypted. For security reasons, the Header does not include information about the encryption algorithm or encryption key. It is assumed in a single-key cryptosystem that the parties exchanging information will know the encryption algorithm and the encryption key.

[25]The purpose of encrypting the **Header:Username** and the **Header:Password** is to avoid sending and storing the information in the clear. It must be stressed, however, that when using a single-key cryptosystem the sending party, the receiving party, or any other party that knows or can discover the encryption algorithm and the encryption key could easily forge, discover, or later manipulate the information.

[26]Applications wishing to use two-key cryptosystems or public key infrastructure should use **SOAP** envelopes and associated toolkits that implement the **W3C XML Signature** specification. In this case, the Header XML must be the first child of the **SOAP** Header element and the Signature element must be the second child to the **SOAP** Header element.

[27]Special credit goes to Amrit Singh Nandrey from E-Filing.com for his offline commentary on this topic.

1.14. Header: MessageIdentifiers

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
MessageIdentifier	1	unbounded

[28] **Header:MessageIdentifiers** is a container element for one or more **Header:MessageIdentifier** elements. **Header:MessageIdentifier** is a unique identifier for a message originally sent. The identifier must be unique to the sending application, but must not be globally unique. In a simple request/response transaction, the **Header:MessageIdentifier** value in the confirmation or response must be the same as the value in the message originally sent. In simple request/response transactions there should only be one **Header:MessageIdentifier** element.

[29] In complex request/response transactions that involve intermediaries, there is a need for multiple **Header:MessageIdentifier** elements so that any application receiving a response can match its unique **Header:MessageIdentifier** with a **Header:MessageIdentifier** in the response. Intermediary applications must, therefore, add a new **Header:MessageIdentifier** element to the Header when forwarding Filing XML. A new **Header:MessageIdentifier** element must be added as the first child of the **Header:MessageIdentifiers** element.

[30] For example, Filing Application 1 ("FA1") creates a Filing Envelope with a **Header:MessageIdentifier** element with, for the purposes of this example, a value equal to 1. FA1 sends the Filing Envelope to Intermediary Application 1 ("IA1"). IA1 forwards the Filing Envelope to Intermediary Application 2 ("IA2"). Finally, IA2 forwards the Filing to a Court's EFM. In this case, IA1 must add a new **Header:MessageIdentifier** element. For purposes of this example, assume the value is 2 (but note that the requirement is for a value unique to the sending application, not a sequential value or a globally unique value). Similarly, IA2 must create a new **Header:MessageIdentifier** with a unique value, 3, for purposes of this example. When the Filing XML reaches the Court's EFM, it will have three **Header:MessageIdentifier** elements, with values 1, 2, and 3.

[31] Assume that FA1, IA1, and IA2 each added information to the **Header:ReplyTo** element so that each application would receive Confirmation XML on the occurrence of a downstream event, such as acceptance or rejection of the filing. Upon a downstream event, the responding application would send a response, in this case Confirmation XML, to each address in the **Header:ReplyTo** element. The responding application would also create a **Header:MessageIdentifiers** element that mirrors the **Header:MessageIdentifiers** element that it received. That is, building on the example above, the responding application's **Header:MessageIdentifiers** element would have three **Header:MessageIdentifier** children elements with the values 1, 2, and 3. In this way, each application that receives a response would be able to match the response with its own **Header:MessageIdentifier** value.

1.15. Header: MessageIdentifier

Data Type: **xsd:string**

[32] **Header:MessageIdentifier** is a unique identifier for a message originally sent. The identifier must be unique to the sending application, but must not be globally unique. In a simple request/response transaction, the **Header:MessageIdentifier** value in the confirmation or response must be the same as the value in the message originally sent. See **Header:MessageIdentifiers** for complex request/response transactions that involve intermediaries.

1.16. Header: ConfirmationIdentification

Data Type: **xsd:string**

[33] **Header:ConfirmationIdentification** is a unique identifier for a confirmation or response. **Header:ConfirmationIdentification** should not be included in a message originally sent.

1.17. Header: Date

Data Type: **xsd:date**

[34] **Header:Date** is the date on which the Header and its Envelope are created.

1.18. Header: Time

Data Type: **xsd:time**

[35] **Header:Time** is the time on which the Header and its Envelope are created.

1.19. Header: Extensions

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Extension	1	unbounded

[36] **Header:Extensions** is a container element for one or more **Header:Extension** elements.

1.20. Header: Person

Data Type: **Person:Person**

[37] Sending applications may populate the **Header:Person** element with any information in the **Person** Schema. However, sending and receiving applications are only required to process and send names and email addresses. Sending and Receiving applications may process postal addresses and fax numbers, if the application has the ability forward filings in these ways. Sending applications should include email addresses, fax numbers, and postal addresses in the **Header:Person** element only if the application intends for the Filing XML to be sent to the address or fax number.

[38] Filings sent to postal addresses or fax machines may apply a stylesheet to the Filing XML so that the information in the Filing is human-readable.

1.21. Header: Organization

Data Type: **Organization:Organization**

[39] Sending applications may populate the **Header:Organization** element with any information in the **Organization** Schema. However, sending and receiving applications are only required to process and send names and email addresses. Sending and Receiving applications may process postal addresses and fax numbers, if the application has the ability forward filings in these ways. Sending applications should include email addresses, fax numbers, and postal addresses in the **Header:Organization** element only if the application intends for the Filing XML to be sent to the address or fax number.

[40] Filings sent to postal addresses or fax machines may apply a stylesheet to the Filing XML so that the information in the Filing is human-readable.

1.22. Header: Extension

Data Type: **Extension:Extension**

[41] A **Header:Extension** is a generic name/value pair that allows applications to send information not included in the **Header** Schema. See **Extension** Schema.

2. Simple Types

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Extension

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Extension/01/>

3.3. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

3.4. Person

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Person/01/>

4. Change History

4.1. 2003-01-15

Editor: Winchel Vincent

Initial Draft

4.2. 2003-07-21

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.3. 2003-07-21

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Header/01/>

Copied

4.4. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Header/01/>

Copied

4.5. 2003-07-26

Editor: Winchel Vincent

Changed Extension namespace and import location to reflect move of Extension to Building Block folder.

4.6. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Changed Person and Organization namespaces.

4.7. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Header/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Page Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Page/01/>

Schema Prefix

Page

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Page/01/Page.xsd>

Table of Contents

1. Elements

Internal

1.1. Page	-67-
1.2. Key	-68-
1.3. Content	-68-

External

2. Simple Types

3. Imported Schemas

3.1. Attributes	-68-
------------------------	------

4. Change History

4.1. 2003-02-18	-68-
4.2. 2003-03-01	-68-
4.3. 2003-06-14	-68-
4.4. 2003-06-14	-68-
4.5. 2003-06-14	-68-
4.6. 2003-06-16	-68-
4.7. 2003-07-23	-69-
4.8. 2003-07-27	-69-
4.9. 2003-07-29	-69-

1. Elements

1.1. Page: Page

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Key	1	1
Content	1	1

[1] **Page** is the intended root element of the schema.

1.2. Page: Key

Data Type: **xsd:string**

[2] **Page:Key** is a unique identifier for the page.

1.3. Page: Content

Data Type: **xsd:string**

[3] **Page:Content** is the encoded document page as a text node. See **Document** Schema.

2. Simple Types

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-03-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/02/Page/01/>

4.3. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Test01/Page/01/>

Copied. Removed mixed content from Page. Removed blank enumeration from Numbers.

4.4. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test01/Page/01/>

Copied

4.5. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test02/Page/01/>

Copied

4.6. 2003-06-16

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Eliminated Number and Link attributes and Numbers simpleType.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Page/01/>

Copied

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.9. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Page/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Payment Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Payment/01/>

Schema Prefix

Payment

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Payment/01/Payment.xsd>

Table of Contents

1. Elements

Internal

1.1. Payment	-70-
1.2. Amount	-70-
1.3. Total	-71-
1.4. NotToExceed	-71-

External

1.5. CreditCard	-71-
------------------------	------

2. Simple Types

3. Imported Schemas

3.1. Attributes	-71-
3.2. CreditCard	-71-

4. Change History

4.1. 2003-07-23	-71-
4.2. 2003-07-27	-71-
4.3. 2003-07-29	-71-

1. Elements

1.1. Payment: Payment

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
CreditCard	1	1
Amount	1	1

[1] **Payment** contains the court payment information, necessary for a vendor or other collection agency to pay the court for filing fees or other amounts collected on the court's behalf. Payment information is not intended to be directly provided to filers unless the court gives permission.

1.2. Payment: Amount

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
Total	1	1
NotToExceed	1	1

[2] **Payment:Amount** is the court's bank account number.

1.3. Payment: Total

Data Type: **xsd:string**

[3] **Payment:AmountNotToExceed** is the routing number for the court's bank.

1.4. Payment: NotToExceed

Data Type: **xsd:string**

[4] **Payment:BankName** is the name of the court's bank.

1.5. Payment: CreditCard

Data Type: **CreditCard:CreditCard**

[5] **Payment:CreditCardNumber** is the name of the court's bank account.

2. Simple Types

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. CreditCard

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/CreditCard/01/>

4. Change History

4.1. 2003-07-23

Editor: Winchel Vincent

Created

4.2. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.3. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Payment/01/>

Copied



ADMINISTRATIVE OFFICE
OF THE COURTS
INFORMATION SERVICES DIVISION

Service Schema

Last Updated: 2003-07-29

Schema Namespace

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Service/01/>

Schema Prefix

Service

Schema Repository Location

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test03/Service/01/Service.xsd>

Table of Contents

1. Elements

Internal

1.1. Service	-73-
1.2. Date	-73-
1.3. Time	-73-
1.4. Type	-73-
1.5. ProcessServerIdentifier	-73-
1.6. ProcessServer	-73-
1.7. Recipient	-73-
1.8. Location	-74-
1.9. Status	-74-

External

1.10. Person	-74-
1.11. Organization	-74-

2. Simple Types

2.1. Statuses	-74-
---------------	------

3. Imported Schemas

3.1. Attributes	-74-
3.2. Organization	-74-
3.3. Person	-74-

4. Change History

4.1. 2003-02-18	-75-
4.2. 2003-03-01	-75-
4.3. 2003-06-14	-75-
4.4. 2003-06-14	-75-
4.5. 2003-06-14	-75-
4.6. 2003-06-16	-75-
4.7. 2003-07-23	-75-
4.8. 2003-07-27	-75-
4.9. 2003-07-29	-75-

1. Elements

1.1. Service: Service

Content Model: sequence

ChildElement(s)	minOccurs	maxOccurs
Date	1	1
Time	0	1
Type	0	1
ProcessServerIdentifier	0	1
ProcessServer	1	1
Recipient	1	1
Status	0	1

[1] **Service** is the intended root element of the schema. **Service** contains information about process of service.

1.2. Service: Date

Data Type: **xsd:date**

[2] **Service:Date** is the date that the process server accomplished service. **Service:Date** is not the **Filing:Date**.

1.3. Service: Time

Data Type: **xsd:time**

[3] **Service:Time** is the time that the process server accomplished service. **Service:Time** is not the **Filing:Time**.

1.4. Service: Type

Data Type: **xsd:string**

[4] **Service:Type** is the type of service accomplished.

1.5. Service: ProcessServerIdentifier

Data Type: **xsd:string**

[5] **Service:ProcessServerIdentifier** is a unique identifier for license number of a process server.

1.6. Service: ProcessServer

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
Person	1	1
Organization	1	1

[6] **Service:ProcessServer** includes name and contact details of a process server.

1.7. Service: Recipient

Content Model: choice

ChildElement(s)	minOccurs	maxOccurs
Person	1	1

Organization	1	1
Location	1	1

[7] **Service:Recipient** is a person, organization, or location where process of service was accomplished.

1.8. Service: Location

Data Type: **xsd:string**

[8] **Service:Location** is the physical location where service was accomplished if not personally served. **Service:Location** could also be a newspaper or publication if service is done by publication. **Service:Recipient** is a required element. If service has failed or is pending, then **Service:Location** should include the location of attempted or failed service or should contain the string "None".

1.9. Service: Status

Data Type: **Statuses**

[9] **Service:Status** is the status of the service attempt. Acceptable values are "Success", "Failure", "Pending", and "Unknown".

1.10. Service: Person

Data Type: **Person:Person**

[10] **Service:Person** is a person involved in service of process.

1.11. Service: Organization

Data Type: **Organization:Organization**

[11] **Service:Organization** is an organization involved in service of process.

2. Simple Types

2.1. Statuses

Data Type: **xsd:string**

Enumeration(s)
Success
Failure
Pending
Unknown

3. Imported Schemas

3.1. Attributes

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Attributes/01/>

3.2. Organization

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Organization/01/>

3.3. Person

<http://www.xmllegal.org/Schema/Court/US/California/2GEFS/BuildingBlocks/Primitives/Person/01/>

4. Change History

4.1. 2003-02-18

Editor: Winchel Vincent

Added Structured Copyright and Update History

4.2. 2003-03-01

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/02/Service/01/>

Added ProcessServerIdentifier element. Changed Time element to be optional.

4.3. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Test01/Service/01/>

Copied. Eliminated blank enumeration from Statuses. Changed Organization and Person references to Typed.

4.4. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test01/Service/01/>

Copied

4.5. 2003-06-14

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/Filing/Tests/Filing/Test02/Service/01/>

Copied

4.6. 2003-06-16

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9.

4.7. 2003-07-23

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test01/Service/01/>

Copied. Eliminated Date.xsd and Time.xsd and included xsd:date and xsd:time.

4.8. 2003-07-27

Editor: Winchel Vincent

Normalized using xmlLegal Normalizer 0.0.9. Changed Person and Organization namespaces.

4.9. 2003-07-29

Editor: Winchel Vincent

Copied From: <http://www.xmllegal.org/Schema/Court/US/California/2GEFS/Filing/Test02/Service/01/>

Copied

2.0 Legal Notices

Unless otherwise agreed, All Rights Reserved except those granted by xmlLegal General Public License at:

<http://www.xmllegal.org/Legal/GeneralPublicLicense.htm>

LICENSED WORKS ARE PROVIDED "AS IS," AND HOLDERS OF INTELLECTUAL PROPERTY RIGHTS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE LICENSED WORKS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

HOLDERS OF INTELLECTUAL PROPERTY WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE LICENSED WORKS.

Contact Winchel "Todd" Vincent III (winchel@mindspring.com) or XMLLegal Help (help@xmllegal.org) for more information